PHYMEM: RETRIEVED PREDICTIVE MEMORIES FOR PHYSICALLY PLAUSIBLE VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-video diffusion models can synthesize visually compelling clips, yet they remain unreliable when prompts require coherent motion, causal interactions, or physically plausible temporal evolution. A key limitation is that text specifies *what* should happen but provides little evidence about *how* the scene should physically unfold. We study whether predictive video representations retrieved from real videos can provide external guidance for physically plausible generation with a frozen Transformer-based latent video diffusion backbone. We introduce PHYMEM, a plug-in memory framework that (i) retrieves prompt-relevant reference clips, (ii) instantiates world-model memories with predictive representations extracted by a frozen V-JEPA2 encoder, (iii) compresses the resulting spatiotemporal tokens into compact memory tokens, and (iv) injects them into the self-attention layers of a frozen video diffusion backbone through a lightweight trainable memory module, without finetuning the backbone. Training updates only a lightweight memory interface, with 33.9M trainable parameters and approximately 0.8K training videos, while the video diffusion backbone, text encoder, and generator VAE remain frozen. Across physics-focused evaluations, PHYMEM improves physical plausibility and physical commonsense under matched inference settings while largely preserving semantic adherence. Diagnostic ablations further indicate that the gains depend on representation source, retrieval relevance, and retrieval budget: V-JEPA2 memory outperforms native VAE memory, prompt-relevant references are more effective than shuffled ones, and retrieval performance peaks at an intermediate budget. These results suggest that retrieved predictive memories offer a parameter-efficient route toward improving physical plausibility in video generation.

1 INTRODUCTION

Recent video diffusion models have achieved remarkable progress in visual fidelity, temporal coherence, and text-video alignment (Blattmann et al., 2023b;a; Bar-Tal et al., 2024; Peebles & Xie, 2023). Modern systems such as Latte (Ma et al., 2025), CogVideoX (Yang et al., 2025), and Wan (Wan et al., 2025) scale latent diffusion (Rombach et al., 2022) and diffusion Transformer architectures into large text-to-video generators capable of producing visually striking results. Yet the resulting videos often remain more *visually convincing* than *physically faithful*. Physics-focused evaluations such as PhyGenBench (Meng et al., 2024) and VideoPhy2 (Bansal et al., 2026) reveal persistent failures in fine-grained physical processes and action-centric physical commonsense: inconsistent object interactions, implausible material transformations, unrealistic fluid dynamics, and smooth-looking motion that violates causal constraints. Figure 1 illustrates several representative cases where the native backbone produces visually plausible but physically incorrect temporal evolution.

One important limitation is an information gap: text prompts specify *what* should happen in a scene but provide little evidence about *how* the scene should physically unfold. Existing efforts to bridge this gap broadly fall into two categories. One line improves the generative backbone itself through scaling model capacity, data curation, and architecture design (Blattmann et al., 2023a; Yang et al., 2025; Wan et al., 2025). Another line injects physical structure through simulation-based guidance (Liu et al., 2024), controllable dynamics (Wang et al., 2026), representation alignment (Zhang et al., 2025b), latent physical knowledge (Cao et al., 2025), rule-derived context (Zhang et al., 2025a), or inference-time world-model rewards (Yuan et al., 2026). While effective, these methods often

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

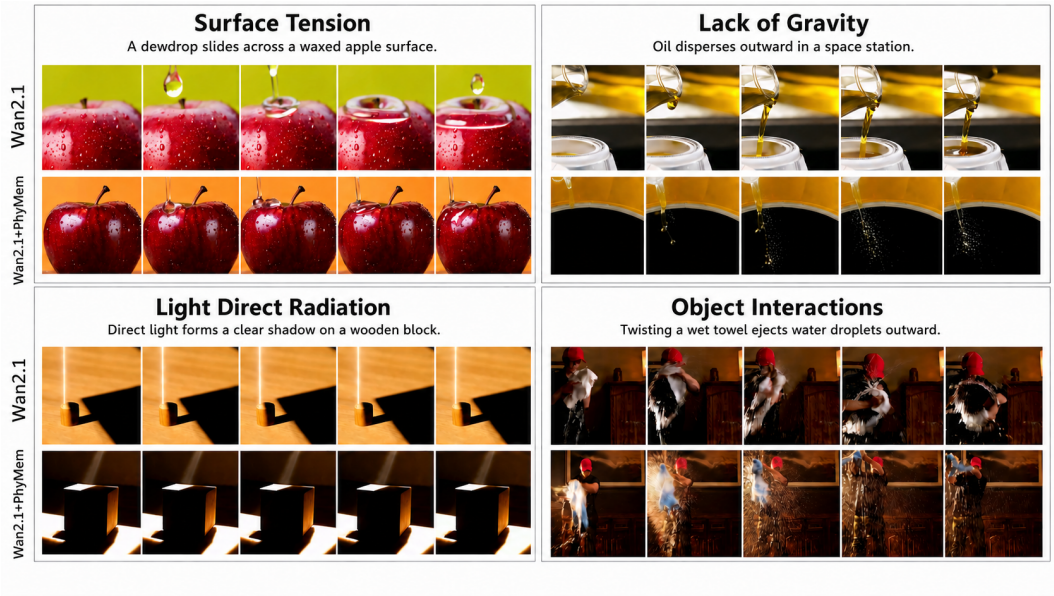


Figure 1: Qualitative comparisons on representative physics-aware prompts. Each case shows five sampled frames from the native Wan2.1 baseline (top) and PHYMEM (bottom). PHYMEM produces more physically plausible temporal evolution, including coherent gravitational motion, more realistic light propagation and shadow formation, and causally consistent object interactions, with retrieved predictive memories from real videos providing additional conditioning to the frozen video diffusion backbone.

require additional generator training, explicit physical assumptions, simulator access, or expensive inference-time optimization.

We ask: *Can retrieved predictive video representations improve the physical plausibility of a frozen video generator through a lightweight memory interface?* Real videos contain examples of scene evolution that may provide useful context beyond text prompts. World-model and JEPA-style representation learning provide a natural source of such memories: these methods learn spatiotemporal representations through prediction (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2023; Assran et al., 2023; Bardes et al., 2025). V-JEPA2 (Assran et al., 2025) scales this paradigm to video understanding, prediction, and planning. Here, "world-model memory" denotes retrieved features from its frozen predictive encoder. Meanwhile, retrieval-augmented generation shows that external examples can provide non-parametric knowledge beyond model parameters (Blattmann et al., 2022; Sheynin et al., 2022; Chen et al., 2022; Song et al., 2025). The remaining challenge is to make retrieved video evidence *usable* by a frozen video diffusion backbone: the reference clips must be encoded with a suitable video representation, compressed into compact memories, and injected without disrupting the pretrained backbone.

We propose PHYMEM, a lightweight memory interface that addresses this challenge. PHYMEM is designed as a plug-in interface for Transformer-based latent video diffusion models with accessible self-attention layers; in this work, we instantiate and evaluate it on Wan2.1. For each text prompt, PHYMEM retrieves prompt-relevant real video clips, encodes them with the frozen V-JEPA2 encoder, compresses the resulting world-model representations into compact memories, and injects the memories into the frozen video diffusion backbone as additional key/value context through a trainable memory module. Training updates only this memory interface (33.9M parameters with approximately 0.8K training videos), while the backbone, text encoder, and generator VAE remain frozen. At inference time, the same memory pathway guides denoising without modifying the native generation path, allowing retrieved predictive memories to provide external temporal guidance.

Our evaluation measures physical plausibility and examines which memory choices affect generation. We compare V-JEPA2 and generator-native VAE representations, prompt-matched and shuffled references, and different retrieval budgets. These diagnostics assess the usefulness of retrieved

context; they do not establish physical-law learning or general physical reasoning. Our contributions are:

- We introduce PHYMEM, a retrieval-based memory interface for frozen Transformer-based video diffusion models. PHYMEM treats real videos as external video evidence and transforms predictive video representations into backbone-compatible key/value memories; we instantiate this interface on Wan2.1.
- We develop a lightweight plug-in memory module that injects retrieved world-model memory tokens into a frozen video diffusion backbone through self-attention, without altering the backbone, text encoder, or generator VAE. Training updates only 33.9M memory-module parameters with approximately 0.8K training videos, while preserving the native no-reference generation path as a matched ablation control.
- We evaluate PHYMEM on PhyGenBench and VideoPhy2 under matched inference settings. Our controlled ablations show that the effectiveness of retrieved memory depends on the representation source, prompt–reference alignment, and retrieval budget: V-JEPA2 memory outperforms Wan VAE memory, matched references outperform shuffled ones, and performance peaks at an intermediate budget.

2 RELATED WORK

Diffusion models for video generation. Video generation builds on diffusion (Ho et al., 2020), latent diffusion (Rombach et al., 2022), and spatiotemporal denoisers (Blattmann et al., 2023b;a; Bar-Tal et al., 2024). Diffusion Transformers replace convolutional denoisers with attention over latent video tokens (Vaswani et al., 2017; Peebles & Xie, 2023), providing scalable backbones for Latte (Ma et al., 2025), CogVideoX (Yang et al., 2025), Open-Sora (Zheng et al., 2024), and Wan2.1 (Wan et al., 2025). These systems improve the generative backbone. We keep Wan2.1 frozen and study retrieved predictive guidance through a lightweight self-attention memory interface.

Retrieval-augmented generation and memory mechanisms. Prior retrieval-augmented diffusion methods condition image generation on retrieved visual or multimodal neighbors (Blattmann et al., 2022; Sheynin et al., 2022; Chen et al., 2022), and retrieval has been applied to temporal generation such as text-driven human motion diffusion (Zhang et al., 2023). The closest related work is DiT-Mem (Song et al., 2025), which learns a plug-and-play memory encoder that maps reference videos into compact tokens inserted into DiT self-attention layers while freezing the diffusion backbone. PHYMEM shares this reference-memory paradigm and studies frozen predictive video representations as a memory source. It uses a pretrained V-JEPA2 encoder and trains a lightweight module to reorganize its features into backbone-compatible key/value memory. We compare with the released DiT-Mem checkpoint and examine representation source, reference relevance, and retrieval budget through diagnostic ablations.

V-JEPA2 world-model representations. World models learn latent dynamics for prediction, imagination, and planning (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2023); foundation models such as Genie (Bruce et al., 2024) extend this line toward interactive environments. JEPA methods learn by latent prediction rather than pixel reconstruction: I-JEPA (Assran et al., 2023) applies this to images, V-JEPA (Bardes et al., 2025) to video, and V-JEPA2 (Assran et al., 2025) scales it to video understanding, prediction, and planning. Our predictive memory uses retrieved features from the frozen V-JEPA2 encoder, which was trained with a joint-embedding predictive objective.

Physics-aware video generation and evaluation. Recent work improves physical plausibility in video generation by adding physical structure beyond text prompts. Simulation- and control-based methods such as PhysGen (Liu et al., 2024) and PhysCtrl (Wang et al., 2026) explicitly recover or control physical trajectories. PhysDiff (Yuan et al., 2023) incorporates physics-based motion priors into human motion diffusion. Latent-knowledge and rule-guided methods inject physical constraints through external knowledge or LLM-derived rules (Cao et al., 2025; Zhang et al., 2025a). Motion-prompting approaches (Geng et al., 2024) use trajectory-based conditioning for controllable generation. Representation-based methods are most related to our work: WMReward (Yuan et al., 2026) repurposes V-JEPA2 as an inference-time reward for physics alignment, and VideoREPA (Zhang et al., 2025b) transfers physics-aware relations from video foundation models through

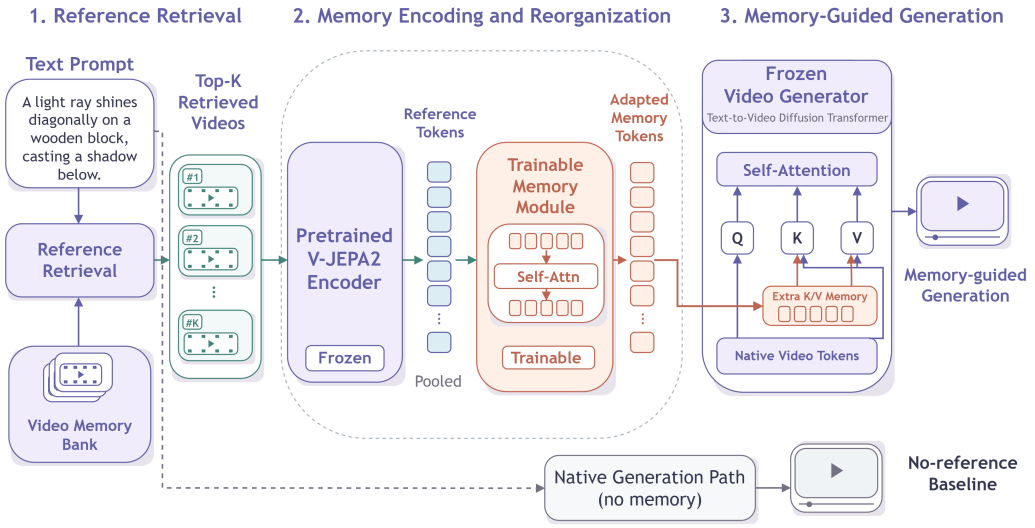


Figure 2: Overview of PHYMEM. For each text prompt, PHYMEM retrieves top-K reference videos from a captioned video bank, encodes them with the frozen V-JEPA2 encoder to obtain world-model token grids, compresses these tokens into compact memories via 3D adaptive pooling, and reorganizes them through a lightweight self-attention memory module. The resulting memory sequence is injected into the self-attention layers of the frozen video diffusion backbone as additional key/value context, providing retrieved predictive guidance throughout denoising.

representation distillation. PHYMEM instead uses retrieved world-model representations as soft temporal priors for a frozen video diffusion backbone, without explicit force parameters, simulator trajectories, inference-time search, or generator finetuning. The interface is broadly intended for latent video diffusion backbones with compatible self-attention layers, while our empirical validation is on Wan2.1. We evaluate on PhyGenBench (Meng et al., 2024) and VideoPhy2 (Bansal et al., 2026), which respectively emphasize fine-grained physical processes and action-centric physical commonsense. Complementary benchmarks such as PhyGenEval (Motamed et al., 2026) and VBench (Huang et al., 2024) further highlight the growing importance of physics-aware evaluation for video generation.

3 METHOD

3.1 OVERVIEW

Given a text prompt p, our goal is to generate a video $\hat{\mathbf{y}}$ that remains semantically aligned with the prompt while exhibiting temporally coherent and physically plausible dynamics. PHYMEM assumes access to a captioned retrieval pool $\mathcal{D}$ of real videos and augments a frozen Transformer-based latent video diffusion backbone with retrieved world-model memories. We instantiate this backbone with Wan2.1. The pipeline proceeds as:

$$p \;\rightarrow\; \{v_i\}_{i=1}^{K} \;\rightarrow\; \{\mathbf{M}_i\}_{i=1}^{K} \;\rightarrow\; \mathbf{R} \;\rightarrow\; \hat{\mathbf{y}}, \tag{1}$$

where $\{v_i\}_{i=1}^{K}$ are the top-K retrieved reference videos, $\mathbf{M}_i \in \mathbb{R}^{N_m \times d_J}$ denotes the set of compressed predictive V-JEPA2 memory tokens extracted from reference v_i, and $\mathbf{R}$ is the reorganized memory sequence injected into the instantiated denoising Transformer. As shown in Figure 2, PHYMEM consists of four stages: (1) reference retrieval, (2) V-JEPA2 tokenization and compression, (3) memory reorganization via self-attention, and (4) key/value memory injection. The instantiated Wan2.1 backbone, text encoder, and Wan VAE remain frozen throughout; only the memory module (parameterized by ψ) is trained.

3.2 Reference Retrieval and V-JEPA2 Memory Extraction

For each prompt p, we retrieve prompt-relevant real videos from $\mathcal{D}$ using dense text retrieval over captions:

$$\{v_i\}_{i=1}^{K} = \mathrm{TopK}_{v \in \mathcal{D}} \; s(\phi_{\text{text}}(p),\, \phi_{\text{text}}(c(v)))\,, \tag{2}$$

where $c(v)$ is the caption of video v, $\phi_{\text{text}}(\cdot)$ is the text embedding model (GTE (Li et al., 2023)), and $s(\cdot,\cdot)$ is the inner product between ℓ_2-normalized embeddings. Retrieval results are precomputed with a FAISS index (Douze et al., 2026); controlled retrieval-budget ablations use prefixes of the same cached top-5 list.

Each retrieved video is then encoded by the frozen stage-1 V-JEPA2 encoder (Bardes et al., 2025; Assran et al., 2025). After uniform frame sampling and V-JEPA2 preprocessing, the encoder produces a raw spatiotemporal token sequence:

$$\tilde{\mathbf{M}}_i = E_{\text{JEPA}}(v_i) \in \mathbb{R}^{N_{\text{raw}} \times d_J}. \tag{3}$$

We reshape $\tilde{\mathbf{M}}_i$ into its spatiotemporal grid and apply parameter-free 3D adaptive average pooling to obtain compact memory tokens:

$$\mathbf{M}_i = \mathrm{Pool}(\tilde{\mathbf{M}}_i) \in \mathbb{R}^{N_m \times d_J}. \tag{4}$$

These pooled tokens are predictive V-JEPA2 memory tokens in the sense defined above: they come from the frozen encoder learned through joint-embedding predictive training, not from invoking the V-JEPA2 predictor during inference. They provide a compact representation of the temporal scene structure observed in the retrieved reference video.

3.3 Memory Reorganization and Key/Value Injection

The memory module f_ψ converts retrieved V-JEPA2 tokens into Wan-compatible key/value memory.

Positional and rank encoding. For each reference, we add a fixed sine-cosine positional encoding $\mathbf{P} \in \mathbb{R}^{N_m \times d_J}$ factorized over the temporal, height, and width axes, together with a learnable retrieval-rank embedding $\mathbf{e}_i \in \mathbb{R}^{d_J}$.

Self-attention refinement. A reference-wise module $\Pi_\psi(\cdot)$, implemented as self-attention blocks followed by normalization and projection, maps the result to the Wan hidden dimension.

Projection and gating. A learned scalar α controls the overall memory strength. During training, a reference has $g_i = 0$ only if dropped; retained references use positive stochastic gates. At inference, valid references use $g_i = 1$. The final memory sequence is:

$$\mathbf{R} = \alpha \; \mathrm{Concat}_{i=1}^{K} \left[g_i \, \Pi_\psi\big(\mathbf{M}_i + \mathbf{P} + \mathbf{1}\mathbf{e}_i^\top\big) \right] \in \mathbb{R}^{(K \cdot N_m) \times d_W}. \tag{5}$$

This formulation keeps reference identity explicit until the final concatenation, so each retrieved clip is refined independently before its tokens enter the shared attention context.

Key/value injection. Let $\mathbf{H} \in \mathbb{R}^{L \times d_W}$ be the native latent-token sequence at a Wan DiT self-attention layer. PhyMem augments keys and values while keeping queries unchanged:

$$\mathbf{Q} = \mathbf{H} W_Q, \qquad \mathbf{K} = [\mathbf{H};\, \mathbf{R}] W_K, \qquad \mathbf{V} = [\mathbf{H};\, \mathbf{R}] W_V, \tag{6}$$

where W_Q, W_K, W_V are frozen Wan attention projections and d_h is the per-head attention dimension. The resulting attention update is:

$$\mathbf{H}' = \mathrm{softmax}\left(\frac{\mathbf{Q}\,\mathbf{K}^\top}{\sqrt{d_h}} \right) \mathbf{V}. \tag{7}$$

Since queries are produced only from $\mathbf{H}$, the output contains updated native video tokens only. Retrieved references expand the attention context but do not propagate as an additional video-token stream, preserving the native path when no references are provided. Because the query stream remains the native latent-token stream, memory can influence denoising through attention context without creating a second generated-token trajectory.

3.4 Training and Inference

Only the memory module parameters ψ are optimized, while the Wan denoising Transformer parameters Θ, text encoder, and Wan VAE remain frozen. We train through the actual memory-augmented Wan2.1 diffusion backbone using the flow-matching objective (Lipman et al., 2022). For a clean video latent $\mathbf{x}_0$ obtained by the frozen VAE encoder and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, we construct the noisy latent:

$$\mathbf{x}_t = (1 - \sigma_t)\, \mathbf{x}_0 + \sigma_t\, \boldsymbol{\epsilon}, \tag{8}$$

and train the memory-augmented backbone to predict the velocity target:

$$\mathcal{L} = \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{x}_0} \left[w(t) \left\| f_{\Theta,\psi}(\mathbf{x}_t,\, t,\, \mathbf{c},\, \mathbf{R}) - (\boldsymbol{\epsilon} - \mathbf{x}_0) \right\|_2^2 \right], \tag{9}$$

where $\mathbf{c}$ denotes text-conditioning features, $w(t)$ is the standard flow-matching timestep weighting, and $f_{\Theta,\psi}$ denotes the frozen Wan denoising Transformer Θ augmented with the trainable memory module ψ. During training, the top-ranked reference is always retained. Lower-ranked references may be dropped, and those retained are softly reweighted to improve robustness to imperfect retrieval (Appendix C.6).

At inference time, PhyMem uses the same retrieval, tokenization, reorganization, and injection pathway but disables stochastic reference regularization: all valid retrieved references are enabled deterministically. The Wan denoising Transformer produces output latents that are decoded into video frames by the frozen Wan VAE decoder. The original no-reference Wan path is preserved by setting the memory sequence to empty, providing a matched baseline for controlled evaluation. Implementation details including memory-module configuration, hyperparameters, and generation settings are provided in Appendix C.

4 Experiments

4.1 Experimental Setup

We evaluate PhyMem on two complementary physics-focused benchmarks.

PhyGenBench (Meng et al., 2024) contains 160 prompts targeting fine-grained physical processes across four domains: Mechanics (e.g., gravity, elasticity, stress), Optics (e.g., reflection, refraction), Thermal (e.g., melting, combustion), and Material (e.g., solubility, hardness). Evaluation proceeds in three stages using VLM-based judges: Stage 1 scores single-frame physical plausibility, Stage 2 scores temporal consistency from start/middle/end frames, and Stage 3 scores full-video physical correctness. Each stage produces a score in $[0, 3]$; we normalize all scores to $[0, 1]$. The *Final* score is the official stage-wise aggregate, and *Cat. Macro* is the unweighted average across the four physical domains. We additionally report the *non-tie (NT) win rate*: the percentage of prompts where one method is judged better than the other, excluding ties.

VideoPhy2 (Bansal et al., 2026) evaluates action-centric physical commonsense on a fixed balanced 120-prompt subset of its official test set. It reports two complementary axes: *Semantic Adherence (SA)*, measuring how well the video matches the text prompt (scale 1–5), and *Physical Commonsense (PC)*, measuring whether the depicted actions obey physical laws (scale 1–5). The *Joint Score* is the percentage of samples satisfying both SA$\geq$4 and PC$\geq$4 simultaneously, capturing videos that are both semantically faithful and physically plausible.

For all controlled experiments, we strictly match prompts, random seeds, video length (33 frames at 832×480), sampling steps (40), guidance scale (5.0), and inference settings between the baseline and our method. We denote the native no-reference Wan2.1 generation path as Wan2.1, and the same frozen Wan2.1 backbone augmented with PhyMem as Wan2.1+PhyMem. Only the PhyMem memory module (33.9M parameters) is trained; the Wan2.1 backbone, text encoder, and Wan VAE remain frozen throughout. Retrieval results are precomputed and reused across experiments, while predictive V-JEPA2 reference features are cached for V-JEPA2-based variants to ensure reproducibility. Appendix B provides further details on evaluator backends, fixed evaluation subsets, and retrieval-bank construction. Exact token counts, grid sizes, and hidden dimensions are reported in Appendix C.

Table 1: Broader contextual comparison on PhyGenBench four-domain diagnostic scores. Each domain score (Mechanics, Optics, Thermal, Material) is the average VLM-judged physical plausibility across prompts in that domain, normalized to $[0, 1]$ (higher is better). Cat. Macro is the unweighted mean of the four domain scores. Context rows use heterogeneous model-specific settings and serve as reference points only; bold marks the stronger value within the controlled reproduced Wan2.1 pair.

Group	Model	Size	Mechanics	Optics	Thermal	Material	Cat. Macro
Closed-source context	Pika (Pika, 2023)	N/A	0.35	0.56	0.43	0.39	0.44
	Gen-3 (Runway, 2024)	N/A	0.45	0.57	0.49	0.51	0.51
	Kling (Kuaishou Technology, 2024)	N/A	0.45	0.58	0.50	0.40	0.49
Open-source context	CogVideoX (Yang et al., 2025)	2B	0.38	0.43	0.34	0.39	0.39
	CogVideoX	5B	0.39	0.55	0.40	0.42	0.45
	Open-Sora V1.2 (Zheng et al., 2024)	1.1B	0.43	0.50	0.44	0.37	0.44
	LaVie (Wang et al., 2024)	860M	0.30	0.44	0.38	0.32	0.36
	Vchitect 2.0 (Fan et al., 2025)	2B	0.41	0.56	0.44	0.37	0.45
Reproduced Wan2.1	Wan2.1	1.3B	0.50	0.69	0.32	0.33	0.46
	Wan2.1+PHYMEM	1.3B	**0.63**	**0.71**	**0.34**	**0.42**	**0.52**

Table 2: Controlled main results under matched inference settings. PhyGenBench scores are normalized VLM-judged stage scores, with Final denoting the official aggregate and NT Win the non-tie pairwise win rate. VideoPhy2 reports SA/PC mean scores and threshold pass rates; Joint requires both SA≥ 4 and PC≥ 4. Bold marks the better value, with ties bolded for both rows.

	PhyGenBench					VideoPhy2				
Method	Stage1	Stage2	Stage3	Final	NT Win	SA	PC	SA≥ 4	PC≥ 4	Joint
Wan2.1	0.45	0.39	0.61	0.48	35.2%	**3.04**	3.58	**24.2%**	51.7%	15.8%
Wan2.1+PHYMEM	**0.51**	**0.45**	**0.69**	**0.55**	**64.8%**	3.02	**3.73**	**24.2%**	**61.7%**	**20.8%**

4.2 EXPERIMENTAL RESULTS

4.2.1 OVERALL PERFORMANCE

Table 1 places our reproduced Wan2.1 results alongside closed-source and open-source models on PhyGenBench. Because external rows use model-specific settings, they provide context rather than a controlled comparison. Within the reproduced Wan2.1 pair, Wan2.1+PHYMEM improves all four domains and the category macro score, with the largest gain in Mechanics.

4.2.2 CONTROLLED PHYSICS EVALUATION

Table 2 reports the main controlled results. With prompts, seeds, video length, and inference settings matched, Wan2.1+PHYMEM raises PhyGenBench Final from 0.48 to 0.55 and wins 64.8% of non-tied comparisons. Gains across all three evaluation stages suggest improved single-frame plausibility and temporal physical consistency.

On VideoPhy2, Wan2.1+PHYMEM improves physical commonsense and Joint Score while preserving the SA pass rate. Mean SA changes only slightly from 3.04 to 3.02, so the physical gain comes with little change in semantic adherence.

A small human preference evaluation on 30 VideoPhy2 pairs yielded 35 votes for PHYMEM, 22 for Wan2.1, and 33 ties, showing a preference toward PHYMEM alongside frequent ties (Appendix B.5).

Efficiency comparison with DiT-Mem. On the same 80-prompt PhyGenBench subset and evaluation protocol, the released DiT-Mem Wan2.1-1.3B checkpoint (Song et al., 2025) scores 0.5333 Final versus 0.5500 for PHYMEM. DiT-Mem uses 10K memory-training videos versus 0.8K; its memory checkpoint contains 78.8M parameters versus our 33.9M. Figure 3 shows similar scores with $12.5\times$ fewer task-specific training videos and approximately $2.3\times$ fewer trainable parameters. This excludes V-JEPA2 pretraining and does not imply lower total compute or faster inference, which also requires retrieval and reference encoding. DiT-Mem retains its original training recipe, making this a contextual checkpoint comparison, not matched retraining.

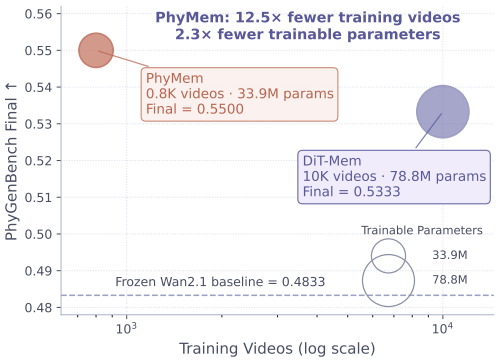


Figure 3: Efficiency on 80 PhyGenBench prompts: training videos (log scale), Final score, and trainable parameters (bubble area). Dashed line: native Wan2.1. DiT-Mem uses its original training recipe (contextual comparison).

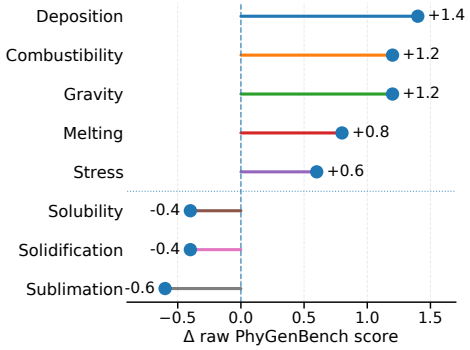


Figure 4: Fine-grained PhyGenBench score changes relative to the native Wan2.1 baseline. The five largest gains and three largest drops are shown.

Table 3: Ablation study examining memory source and reference relevance on a fixed 80-prompt PhyGenBench diagnostic subset. Domain scores and Cat. Macro follow the four-domain view in Table 1 (normalized to $[0, 1]$, higher = better). All rows use the same diagnostic prompts, settings, and evaluator; they are independent from the main results in Table 2. Bold marks the best value within each ablation block and metric column.

Block	Variant	K	Mechanics	Optics	Thermal	Material	Cat. Macro
Memory source	Wan VAE memory	3	0.58	0.67	0.42	**0.38**	0.51
	V-JEPA2 memory	3	**0.63**	**0.77**	**0.51**	0.30	**0.55**
Reference relevance	No references	0	0.49	0.65	0.31	**0.35**	0.45
	Shuffled references	3	0.53	0.65	0.38	0.32	0.47
	Top-K references	3	**0.63**	**0.77**	**0.51**	0.30	**0.55**

Figure 4 shows the largest PhyGenBench gains on deposition, combustibility, gravity, melting, and stress. These results identify where PHYMEM improves most in the evaluated setting, without establishing which reference features account for the gains.

The largest drops occur in sublimation, solidification, and solubility. Mismatches between retrieved references and target dynamics are one possible explanation, but the category scores alone do not establish this cause. Full subcategory results are provided in Appendix B.4.

4.2.3 ABLATION STUDY

We conduct three controlled ablations to examine how the memory source, prompt–reference alignment, and retrieval budget affect generation. Table 3 summarizes these findings on a fixed 80-prompt PhyGenBench diagnostic subset. We have the following observations.

Predictive V-JEPA2 memory outperforms Wan VAE memory. The memory-source block compares predictive V-JEPA2 memory with Wan VAE memory under the same retrieval setting (K=3). The Wan VAE memory baseline uses a generator-native VAE latent construction: retrieved reference videos are encoded by the native Wan VAE encoder and injected in place of V-JEPA2 representations learned through joint-embedding predictive training (construction details in Appendix C.7). Predictive V-JEPA2 memory obtains a $+7.8\%$ higher category macro score (0.55 vs. 0.51), with particularly large gains in Mechanics ($+8.6\%$, $0.58 \rightarrow 0.63$), Optics ($+14.9\%$, $0.67 \rightarrow 0.77$), and Thermal ($+21.4\%$, $0.42 \rightarrow 0.51$). This result is consistent with our hypothesis that predictive video representations provide a more useful memory source than reconstruction-oriented VAE latents in this setting. However, the encoders also differ in architecture, pretraining, and input projection, so this comparison does not isolate the effect of the predictive objective.

Prompt-relevant references are critical. Shuffled retrieval randomly permutes prompt-to-reference-list assignments while preserving reference count, memory format, and injection mechanism (Appendix C.5). This breaks semantic alignment without changing memory length or the overall feature distribution. The control tests whether the gain depends on prompt-specific reference information when the amount and form of injected memory remain fixed. Top-K retrieval outperforms shuffled retrieval (0.55 vs. 0.47, $+17.0\%$ relative) and no references (0.55 vs. 0.45, $+22.2\%$ relative). Shuffling yields only a $+4.4\%$ gain over no references (0.47 vs. 0.45), supporting the importance of prompt-aligned context beyond additional key/value capacity.

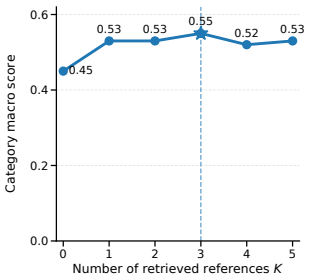


Figure 5: Retrieval-budget ablation. The category macro score peaks at $K{=}3$, indicating that retrieved memory requires calibration rather than monotonic scaling.

Retrieval relevance audit. We manually audited the top-3 references for 20 prompts, counting a reference as physically relevant when it depicts the same or a closely related physical process, beyond shared objects or scenes. Of these prompts, 13 (65.0%) have at least one relevant reference. This small audit shows that semantic retrieval can surface related physical processes, but does not measure their downstream contribution.

Retrieval budget. Figure 5 visualizes how generation changes with the retrieval budget. The $K{=}1, \ldots, 5$ variants use prefixes of the same saved top-5 retrieval list, so the sweep changes only the number of active reference slots while keeping the retrieval backend and ranking fixed. Performance improves up to $K{=}3$ but drops when additional references are injected. This trend is consistent with lower-ranked references introducing redundant or conflicting context and favors an intermediate retrieval budget in this setting. It does not establish which information in the references drives the improvement.

5 Discussion and Limitations

PhyMem supplies predictive video features as attention context without simulating or verifying physical dynamics. Our results do not disentangle semantic cues, dynamic information, and event-template transfer, or establish general physical reasoning.

Several limitations follow from this design. First, memory guidance remains constrained by the frozen backbone's capacity, and failures persist (Appendix A.3). Second, the non-monotonic budget curve is consistent with sensitivity to additional references, which may introduce irrelevant or conflicting context. Third, the interface targets Transformer-based latent video diffusion models with compatible self-attention layers, but our experiments validate only Wan2.1; cross-backbone generality remains untested. Finally, our main evidence relies on VLM assessments, which may miss physical violations or reflect visual appearance. The small human preference evaluation provides supplementary evidence, but combines action fidelity, physical plausibility, and realism; larger studies and broader reasoning tasks remain needed.

6 Conclusion

PhyMem converts frozen V-JEPA2 features into key/value memory for frozen Wan2.1 through a lightweight trainable interface. Under matched inference settings, it improves physical plausibility and commonsense on two benchmarks while largely preserving semantic adherence. Ablations show that effectiveness depends on representation source, prompt–reference alignment, and retrieval budget, supporting predictive memory as a parameter-efficient approach in this setting. The relative contributions of semantic cues, dynamic information, and event-template transfer remain open. Anonymized code: `https://anonymous.4open.science/r/phymem2026/`.

AI USE STATEMENT

The core research idea and proposed method in this work were conceived and developed by the authors. Generative AI assistants were used in an assistive role to provide feedback on methodology and experimental design, refine research hypotheses and their scope, and support the interpretation of experimental results. We also used these tools for literature discovery and search, manuscript drafting and revision, reference formatting, research code development and debugging, visualization, and the creation and refinement of scientific figures. AI-assisted outputs were reviewed by the authors; experimental claims were checked against experimental records and source code, and cited references were verified against original sources where applicable.

Separately, vision-language models are used as automated evaluators in some benchmark evaluations, as described in the corresponding evaluation protocols. These models are part of the experimental evaluation pipeline rather than assistance in manuscript preparation.

The authors take responsibility for the final content of this work, including the accuracy of its claims, reported results, citations, code, and AI-assisted text and artifacts.

ETHICS STATEMENT

Improving the plausibility of generated videos can support creative and educational applications, but may also make misleading synthetic content more convincing. Generated clips should be identified as synthetic, and physical plausibility should not be treated as evidence of factual accuracy or suitability for safety-critical decisions. Our use of existing video datasets and pretrained models inherits potential biases and privacy concerns; reuse and redistribution should respect the original data and model terms and the rights of depicted individuals. The small human preference evaluation is reported through aggregate judgments without annotator identifiers and should be interpreted within its stated scope and limitations.

REPRODUCIBILITY STATEMENT

Section 3 describes the memory interface and training procedure, and Section 4 specifies the evaluation setup. Appendix B provides benchmark details and the human preference protocol. Appendix C documents retrieval-bank construction, data splits, training and generation settings, compute estimates, memory configuration, and ablation protocols. The controlled comparisons use matched inference settings and cached retrieval results as described in these sections. An anonymized code link is provided in the Conclusion. The DiT-Mem comparison retains the released checkpoint's original training recipe and is identified as contextual rather than fully matched retraining.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3680528.3687614.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2025. ICLR 2025 journal track.

Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Semi-parametric neural image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575, 2023b.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 4603–4623, 2024.

Qinglong Cao, Xirui Li, Ding Wang, Chao Ma, Yuntian Chen, and Xiaokang Yang. Latent knowledge-guided video diffusion for scientific phenomena generation from a single initial frame, 2025.

Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W. Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, et al. The faiss library. *IEEE Transactions on Big Data*, 12(2):346–361, 2026. doi: 10.1109/TBDATA.2025.3618474.

Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, et al. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. *arXiv preprint arXiv:2501.08453*, 2025. doi: 10.48550/arXiv.2501.08453.

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Fitsum Reda, Simon Niklaus, Michael Rubinstein, Deqing Sun, and Jonathan T. Barron. Motion prompting: Controlling video generation with motion trajectories. *arXiv preprint arXiv:2412.02700*, 2024.

David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Kuaishou Technology. Kling ai, 2024. URL `https://klingai.kuaishou.com/`.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision (ECCV)*, pp. 360–378, 2024.

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, et al. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 948–958, March 2026.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, et al. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

Pika. Pika 1.0, 2023. URL `https://pika.art/`.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Runway. Runway gen-3 alpha: Ai video generation with temporal control, 2024. URL `https://runway.com/research/introducing-gen-3-alpha`.

Shelly Sheynin, Oron Ashual, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

Selena Song, Ziming Xu, Zijun Zhang, Kun Zhou, Jiaxian Guo, Lianhui Qin, and Biwei Huang. Learning plug-and-play memory for guiding video diffusion models. *arXiv preprint arXiv:2511.19229*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, et al. Wan: Open and advanced large-scale video generative models, 2025.

Chen Wang, Chuhao Chen, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie Liu. Physctrl: Generative physics for controllable and physics-grounded video generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, et al. Lavie: High-quality video generation with cascaded latent diffusion models, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, et al. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16010–16021, 2023.

Ke Zhang, Cihan Xiao, Jiacong Xu, Yiqun Mei, and Vishal M. Patel. Think before you diffuse: Infusing physical rules into video diffusion, 2025a.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 364–373, 2023.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, et al. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

A Additional Qualitative Analysis

A.1 Additional Qualitative Comparisons

Figure A1 provides additional qualitative comparisons beyond the representative cases shown in Figure 1. Across these prompts, PhyMem more consistently preserves physically plausible temporal evolution, while the native baseline more often exhibits weaker interaction structure, appearance drift, or less convincing material response.

A.2 Retrieval-Augmented Generation Examples

Figure A2 visualizes representative retrieval-augmented generation examples. For each prompt, we show sampled generated frames and the top-5 retrieved reference videos used by PhyMem.

A.3 Residual Failure Cases

Figure A3 presents representative residual failure cases. While PhyMem often improves temporal evolution relative to the native baseline, it does not guarantee physically correct outputs. In the Tyndall-effect example, PhyMem generates a more plausible dust-visible light path, whereas the baseline resembles a flame-like jet rather than light scattering through dust. Even so, the beam is still not perfectly aligned with the flashlight source. In the hardness example, neither method clearly shows the egg fracturing on impact. However, PhyMem conveys a more rigid collision through contact response and dust, while the baseline appears more like a lightweight deformable object and exhibits stronger appearance drift. These cases highlight an important limitation of the method: retrieved world-model memory can steer generation toward more plausible dynamics, but it cannot fully correct failures of the frozen backbone or enforce physics as a hard constraint.

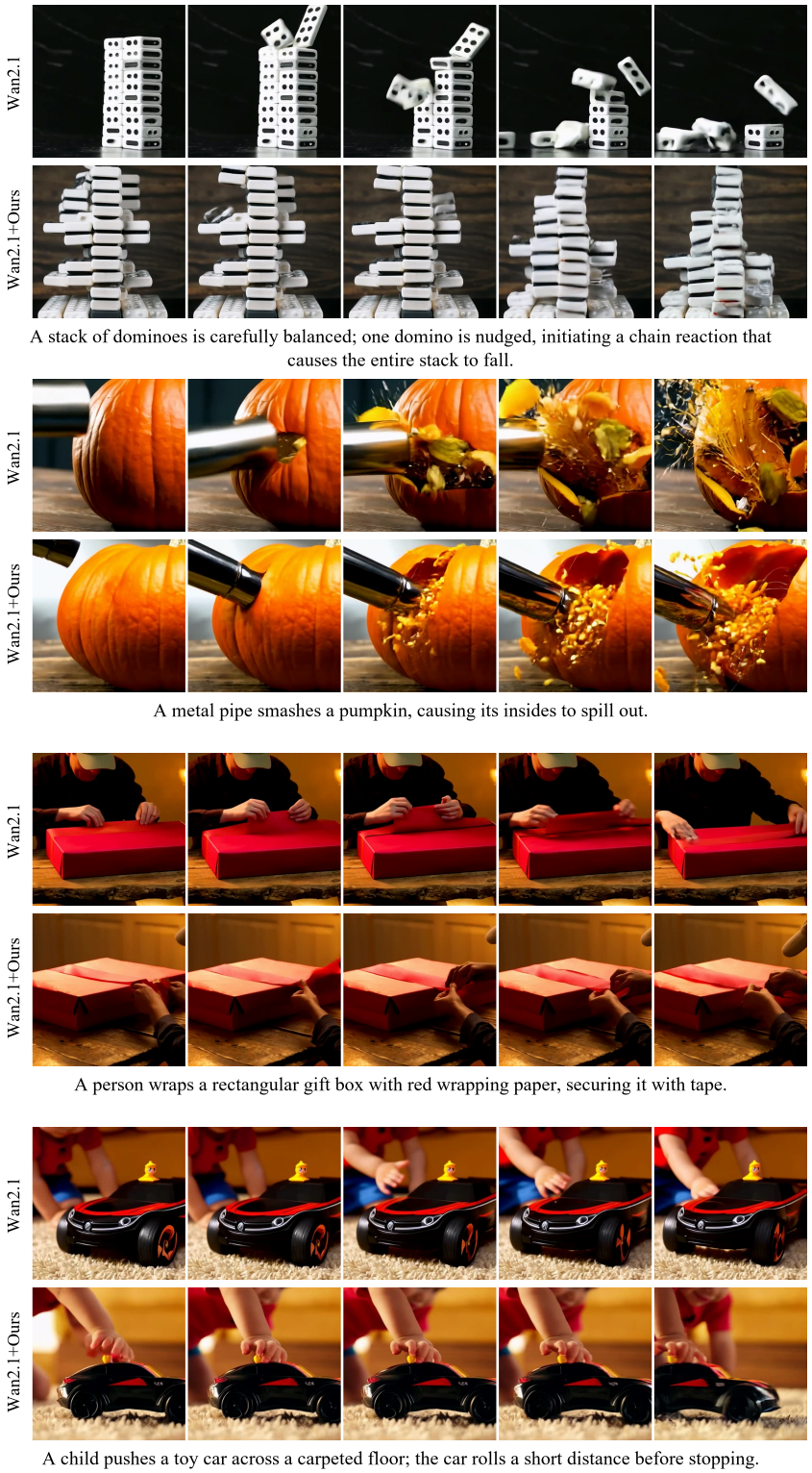


Figure A1: Additional qualitative comparisons on representative physics-aware prompts. Each example shows sampled frames from the native Wan2.1 baseline and Wan2.1+PHYMEM under matched inference settings, illustrating improved temporal evolution and physical plausibility across diverse scene types.

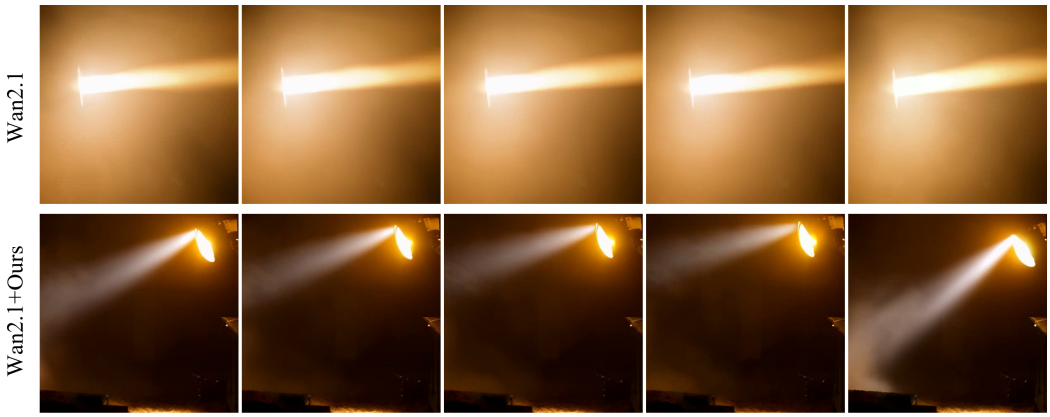


A black pen is used to write on the smooth, white surface of a notebook, showcasing the interaction between the pen and the notebook surface.

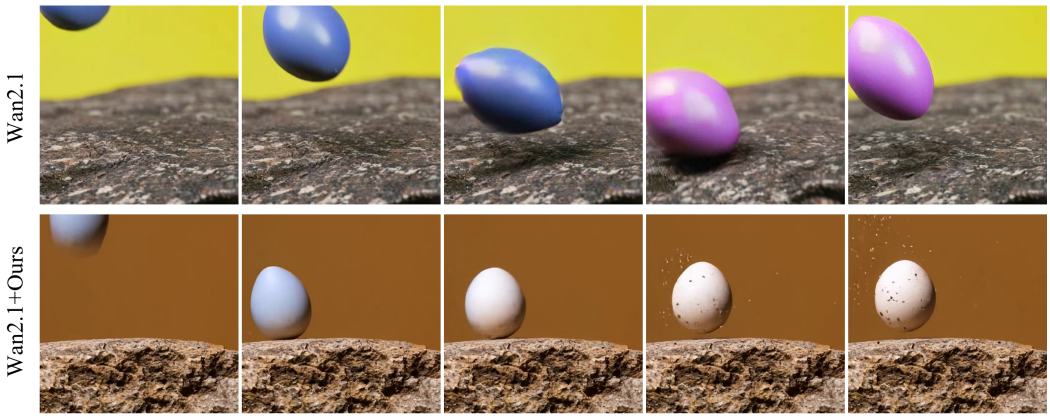


A clear glass of honey is gently poured into a glass of milk.

Figure A2: Retrieval-augmented generation examples. For each prompt, the first row shows sampled generated frames, and the second row shows the top-5 retrieved reference videos used by PHYMEM. These examples visualize the real-video evidence encoded into retrieved world-model memories.



A ray of light generated by a flashlight is passing through a dusty room.



A delicate, fragile egg is hurled with significant force towards a rugged, solid rock surface, where it collides upon impact

Figure A3: Residual failure cases. Top: flashlight beam in a dusty room. PHYMEM better captures a Tyndall-effect light path, but the beam origin remains imperfect. Bottom: egg colliding with a rock. PHYMEM produces a stronger rigid-impact impression with dust, but the egg still does not visibly fracture.

B Additional Experimental Details

B.1 Benchmark Protocol Details

PhyGenBench (Meng et al., 2024) contains 160 prompts designed to evaluate fine-grained physical laws in text-to-video generation. The prompts are commonly grouped into four domains: *Mechanics*, *Optics*, *Thermal*, and *Material Properties*. The benchmark reports three stage-wise scores in $[0, 3]$: Stage 1 evaluates physical plausibility from sampled single frames, Stage 2 evaluates temporal consistency using start/middle/end frames, and Stage 3 evaluates full videos with a VLM-based judge. Unless otherwise stated, we normalize PhyGenBench scores to $[0, 1]$.

VideoPhy2 (Bansal et al., 2026) provides a complementary evaluation focused on action-centric physical commonsense and semantic adherence. Its official protocol reports semantic adherence (SA), physical commonsense (PC), and Joint Score, defined as the percentage of samples satisfying both $\mathrm{SA} \geq 4$ and $\mathrm{PC} \geq 4$. We report VideoPhy2-AutoEval results on a fixed balanced 120-prompt subset sampled from the official VideoPhy2 test set, balancing hard/non-hard prompts and activity/interaction categories where possible. We use the benchmark's upsampled captions for generation and evaluation. The subset is fixed using a saved random seed and archived prompt IDs to ensure reproducibility across model variants.

B.2 Representative Benchmark Prompts

Table A1 gives representative prompts from the two evaluation benchmarks. PhyGenBench prompts specify a physical scenario and expected physical law, while VideoPhy2 prompts describe dynamic actions or interactions that are judged for both semantic adherence and physical commonsense.

Table A1: Representative prompt examples from PhyGenBench and VideoPhy2. These examples are illustrative and are not experimental results.

Benchmark	Prompt example	Physical focus	Evaluation focus
PhyGenBench	A cup of water is slowly poured out in the space station, releasing the liquid into the surrounding area.	Gravity / lack of gravity	Stated law compliance
PhyGenBench	A clear plastic straw is slowly inserted into a glass of crystal-clear water, revealing the visual changes as the straw interacts with the liquid.	Refraction	Stated law compliance
PhyGenBench	A timelapse captures the reaction as concentrated sulfuric acid is poured onto a piece of bread.	Dehydration property	Stated law compliance
VideoPhy2	A golfer addresses the ball, takes a backswing, and hits the ball, sending it arcing high into the air.	Gravity; conservation of mass	SA and PC
VideoPhy2	Multiple balls are grouped near a pocket; a precise shot pockets two of the balls.	Hardness / contact dynamics	SA and PC
VideoPhy2	A person flips their hair over and uses a hairdryer to lift the roots, giving their hair more volume.	Fluid dynamics	SA and PC

B.3 PhyGenBench Evaluation Backend

For our reproduced PhyGenBench evaluations, we use Kimi K2.6 as a fixed evaluator backend following the PhyGenEval prompt structure. The backend, prompts, parsing logic, and normalization are kept fixed across the reproduced Wan2.1 and Wan2.1+PhyMem comparisons and across all PhyGenBench ablations. As a result, the reported conclusions should be interpreted as controlled within-backend comparisons under identical evaluation settings, while broader cross-model context involving externally reported PhyGenBench numbers should be treated as contextual rather than strictly numerically matched.

B.4 PHYGENBENCH SUBCATEGORY ANALYSIS

Figure A4 reports the full PhyGenBench subcategory-level changes between Wan2.1+PHYMEM and the native Wan2.1 baseline. This figure complements the main-text summary in Figure 4.

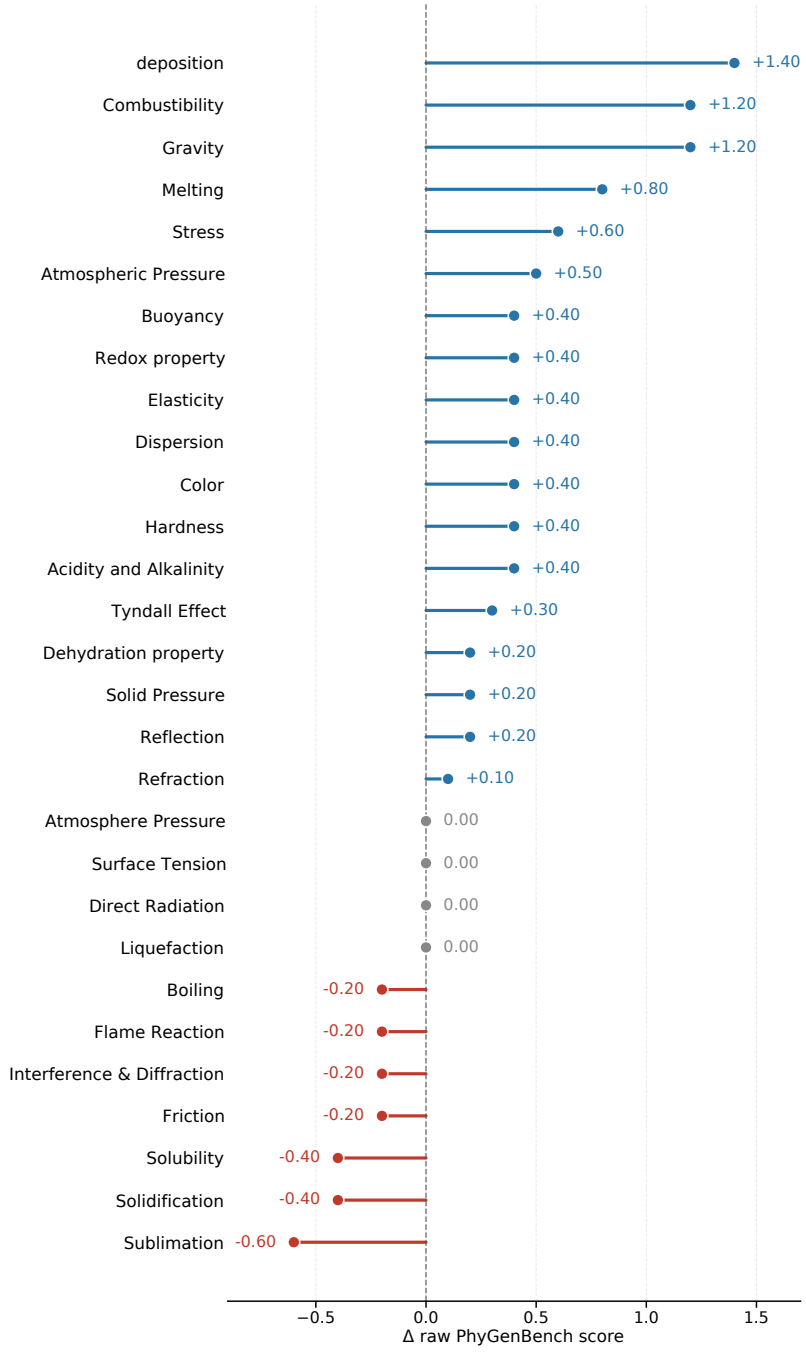


Figure A4: Full PhyGenBench subcategory-level score changes for Wan2.1+PHYMEM relative to the native Wan2.1 baseline. Positive values indicate subcategories where retrieved world-model memory improves physical plausibility; negative values indicate residual failure modes.

B.5 SMALL-SCALE HUMAN PREFERENCE EVALUATION

We evaluated 30 pairs of existing VideoPhy2 generations, comparing native Wan2.1 with PHYMEM. The annotation package sampled pairs across category (object interactions versus sports and physical activities) and difficulty (easy versus hard), and randomized the left/right method assignment. Three annotators each supplied one judgment per pair, yielding 90 recorded votes. The question was: "Which video better depicts the described action with physically plausible and realistic motion?" The available choices were left, right, or tie.

Table A2 reports all votes, including ties, and the majority outcome for each pair. A pair-level outcome requires at least two matching judgments; every pair had a majority in this study. PHYMEM received more preferences overall, but ties were frequent. The difficulty breakdown also shows that this trend was not uniform: easy pairs favored PHYMEM, whereas hard pairs received more votes for Wan2.1 than for PHYMEM.

Table A2: Human preference results on 30 VideoPhy2 pairs with three judgments per pair. Vote-level rows count judgments; the final row counts pair-level majority outcomes. Easy and hard rows partition the 90 votes.

Aggregation	PHYMEM	Wan2.1	Tie
All votes ($n = 90$)	35 (38.9%)	22 (24.4%)	33 (36.7%)
Easy-pair votes ($n = 42$)	25	6	11
Hard-pair votes ($n = 48$)	10	16	22
Pair majority ($n = 30$)	10 (33.3%)	7 (23.3%)	13 (43.3%)

This small evaluation offers descriptive evidence of human preference alongside automated benchmark scores. The question combines action fidelity, physical plausibility, and visual realism, so it is not a direct test of physical-law compliance. Repeated judgments on the same pairs are not independent samples, and these counts do not establish a statistically significant advantage or generalization beyond the evaluated pairs.

C ADDITIONAL IMPLEMENTATION DETAILS

C.1 RETRIEVAL BANK AND TRAINING DATA

We build the retrieval bank and training targets from OpenVidHD, the high-resolution subset of OpenVid-1M (Nan et al., 2025). Our retrieval bank is defined as the full non-static subset of OpenVidHD and contains 115,705 videos. We further construct a fixed target set of 1,000 videos, split into 800 training, 100 validation, and 100 test targets. Captions for the retrieval bank are taken from the corresponding OpenVidHD metadata. We embed both target prompts and retrieval-bank captions using `Alibaba-NLP/gte-base-en-v1.5` (Li et al., 2023), ℓ_2-normalize the embeddings, and perform nearest-neighbor search with a FAISS `IndexFlatIP` index (Douze et al., 2026). For each target prompt, the top-5 retrieved references are precomputed once and reused across training, evaluation, and ablations. This fixed retrieval cache ensures that retrieval-budget comparisons vary only the number of active references: the $K = 1, \ldots, 5$ settings activate prefixes of the same saved top-5 list. Similarly, the shuffled-reference ablation preserves the cached reference lists but permutes their assignment to prompts, isolating the effect of reference relevance without changing memory length or retrieval format.

C.2 TRAINING HYPERPARAMETERS

Table A3 summarizes the training hyperparameters for the PHYMEM checkpoint used in the main results. Training uses the fixed 800-video training split described above; after filtering unavailable target latents, 792 training videos remain. We train only the memory module while keeping the Wan2.1 backbone, text encoder, and Wan VAE frozen. Checkpoints are saved every 20 optimization steps, and the main evaluation uses a fixed checkpoint selected before running the final PhyGenBench and VideoPhy2 experiments.

Table A3: Training hyperparameters for the main PHYMEM checkpoint.

Item	Value
Trainable parameters	≈ 33.9M
Base generator / text encoder / Wan VAE	Frozen
Optimizer	AdamW
Learning rate	1×10^{-4}
Batch size	4
Hardware	3 H100 GPUs
Approximate training time to main checkpoint	Within one day of active training

C.3 GENERATION SETTINGS

Table A4 reports the generation settings used for the controlled main results unless otherwise specified. The Wan2.1 baseline uses the native no-reference Wan2.1 generation path with the same prompts, seeds, video length, resolution, sampler, and guidance scale. Wan2.1+PHYMEM uses the same frozen Wan2.1 backbone and the same sampling settings, with retrieved memories enabled through the selected PHYMEM checkpoint described in Appendix C.2. We do not use Wan prompt expansion; PhyGenBench uses the benchmark prompts, while VideoPhy2 uses the benchmark-provided upsampled captions.

Table A4: Generation settings for the controlled main experiments.

Item	Value
Base model	Wan2.1-T2V-1.3B
Retrieval budget	Top-3 references
Resolution	832×480
Generated frames	33
Output fps	15
Sampling steps	40
Guidance scale	5.0
Random seed	42
Prompt expansion	Disabled

C.4 APPROXIMATE COMPUTE BUDGET

The main PHYMEM checkpoint is trained on 3 H100 GPUs within one day of active training, as summarized in Table A3. For the controlled main experiments, generating one benchmark sweep for a single model variant typically takes a few wall-clock hours under the settings in Table A4, and the corresponding PhyGenBench and VideoPhy2 evaluation passes also complete within a few hours each. These estimates are intended as approximate reproduction guidance rather than a full project-wide compute audit, since the broader research process also included preliminary runs, diagnostics, and discarded checkpoints that are not individually itemized in the paper.

C.5 REFERENCE-RELEVANCE AND BUDGET DIAGNOSTIC PROTOCOL

The reference-relevance and retrieval-budget ablations are evaluated on the same fixed 80-prompt PhyGenBench diagnostic subset used by the other mechanism rows in Table 3. Prompts, generation settings, evaluator backend, and parsing logic are held fixed across all rows. For each prompt, we retrieve and cache a ranked top-5 reference list once. The $K = 1, \ldots, 5$ variants activate prefixes of this saved list, and the $K = 0$ row disables reference memory while preserving the native no-reference generation path. Consequently, the retrieval-budget sweep changes only the number of active reference memories, without changing the retrieval backend or ranking order.

For the shuffled-reference condition, we keep K fixed and preserve the memory tensor shape, token count, reference preprocessing, and injection pathway. We then randomly permute the prompt-to-reference-list assignment within the diagnostic set after retrieval. This preserves the marginal

distribution of reference videos and the amount of injected memory, while breaking semantic alignment between each prompt and its references. The comparison therefore tests whether prompt-aligned retrieved context is important beyond the capacity increase from additional key/value memory.

C.6 Memory-Module Configuration

Table A5: Memory-module configuration used in our experiments. The table instantiates the symbolic quantities used in Section 3, including the raw V-JEPA2 token grid, compressed memory layout, Wan hidden dimension, retrieval budget, and trainable parameter count.

Quantity / component	Default value
V-JEPA2 token dimension d_J	1408
Compressed memory grid	$8 \times 5 \times 5$
Tokens per reference N_m	200
Wan hidden dimension d_W	1536
Memory-module self-attention blocks L_A	2
Maximum references K	5
Global memory scale α	initialized to 0.2
Training g_i	0 if dropped; otherwise sampled from $\mathcal{U}(0.7, 1.0)$
Inference g_i	1 for valid references
Trainable parameters	≈ 33.9M

C.7 Wan VAE Memory Baseline

The Wan VAE memory ablation compares two representation sources within the reference-memory interface. This baseline is a generator-native VAE latent construction: we replace V-JEPA2 reference tokens with latents encoded by the native Wan VAE encoder. For each retrieved reference video, we use the same Wan VAE encoder that defines the Wan generator latent space and encode the reference into a latent tensor with channel dimension 16 and shape $[16, 21, 60, 104]$ after temporal normalization. To match the V-JEPA2 memory budget, we apply 3D adaptive average pooling to a fixed $8 \times 5 \times 5$ grid, flatten the result into 200 tokens, and project each 16-dimensional VAE token to the memory-module input width using a lightweight stem:

$$[16, 21, 60, 104] \to [16, 8, 5, 5] \to [200, 16] \xrightarrow{\text{LayerNorm+Linear}} [200, 1408]. \tag{10}$$

After this source-specific stem, the Wan VAE memory variant uses the same retrieval slots, masking convention, memory-module architecture, and self-attention injection interface as the predictive V-JEPA2-memory variant. The comparison evaluates reconstruction-oriented Wan VAE features against predictive V-JEPA2 features within this interface; differences in encoder architecture, pretraining, and the source-specific stem prevent attributing the result solely to the pretraining objective.

C.8 Caching and Missing-Token Handling

For efficiency, we precompute both Wan VAE latents and V-JEPA2 tokens. The cache used in our experiments contains 5696 latent files with shape $[1, 16, 21, 60, 104]$ and 5696 V-JEPA2 token files with shape $[200, 1408]$. All latent clips are temporally normalized to 21 latent frames, corresponding to approximately 85 raw frames, or about 3.4 seconds at 25 fps. When a reference-token file is unavailable, the corresponding memory slot is zero-filled and marked invalid by the reference-existence mask. This keeps the training batches shape-consistent while ensuring that missing references are not interpreted as valid retrieved evidence.

C.9 Masks, Gates, and Wrapper Behavior

Reference validity is represented by a token-level attention mask. We start from a reference-existence mask in $\{0, 1\}^{B \times K}$ and expand each valid reference across its 200 memory tokens, producing a token-level mask in $\{0, 1\}^{B \times (K \cdot 200)}$. The memory sequence itself always follows a fixed $K \cdot 200$ slot layout. Wan's native spatiotemporal rotary positional encoding, based on RoPE (Su et al., 2024),

is applied only to the main video tokens; retrieved reference tokens remain in the memory-module feature space and are used only as external key/value memory.

During training, we apply reference regularization to improve robustness to imperfect retrieval. The top-ranked reference is always retained, while lower-ranked references are independently dropped with probability 0.3. Active lower-ranked references are softly reweighted by gates sampled uniformly from $[0.7, 1.0)$. For distributed training, dropped references are represented as zeroed memory slots at fixed key/value positions, which preserves static tensor shapes and avoids dynamic sequence lengths. During inference, the wrapper applies a strict token-level attention mask for invalid references. This distinction only affects dropped or missing slots; when all selected reference slots are valid, the memory pathway is deterministic.

Table A6: Training-time and inference-time behavior of the full implementation. The table highlights the implementation differences introduced by reference regularization, distributed training, and inference-time masking.

Aspect	Training	Inference
Wan backbone	Frozen with activation checkpointing	Frozen, eval mode
Reference mask	Slotwise dropout; zeroed slots	Valid refs only; strict token mask
Soft gates	Rank-aware random gates	All valid gates set to 1
Memory-module mode	Train mode, α learnable	Eval mode, α fixed
Injection wrapper	Dedicated training wrapper	Patched Wan blocks